대 한 민 국 특 허 청
KOREAN INTELLECTUAL PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출 원 번 호 : 10-2002-0060295
Application Number

출 원 년 월 일 : 2002년 10월 02일
Date of Application OCT 02, 2002

출 원 인 : 학교법인대우학원
Applicant(s) DAEWOO EDUCATIONAL FOUNDATION

2003 년 07 월 30 일

특 허 청 장
COMMISSIONER

【서지사항】

【서류명】 서지사항 보정서

【수신처】 특허청장

【제출일자】 2002.10.04

【제출인】

【명칭】 학교법인 대우학원

【출원인코드】 2-1999-901351-3

【사건과의 관계】 출원인

【대리인】

【성명】 진천웅

【대리인코드】 9-1998-000533-6

【포괄위임등록번호】 2000-055603-1

【사건의 표시】

【출원번호】 10-2002-0060295

【출원일자】 2002.10.02

【심사청구일자】 2002.10.02

【발명의 명칭】 생물정보학에서의 데이터베이스 처리 방법

【제출원인】

【접수번호】 1-1-02-0325265-21

【접수일자】 2002.10.02

【보정할 서류】 특허출원서

【보정할 사항】

【보정대상항목】 발명자

【보정방법】 정정

【보정내용】

【발명자】

【성명의 국문표기】 김재훈

【성명의 영문표기】 KIM,Jai Hoon

【주민등록번호】 601208-1025524

【우편번호】 449-840

【주소】 경기도 용인시 수지읍 상현리 만현마을 쌍용 1차 701동 1703 호

【국적】 KR

【발명자】
【성명의 국문표기】 김민준
【성명의 영문표기】 KIM,Min Joon
【주민등록번호】 790220-1454734
【우편번호】 442-191
【주소】 경기도 수원시 팔달구 우만1동 519-8 204호
【국적】 KR
【발명자】
【성명의 국문표기】 이성준
【성명의 영문표기】 LEE,Seong Joon
【주민등록번호】 770416-1932122
【우편번호】 690-042
【주소】 제주도 제주시 용담2동 627-15
【국적】 KR
【발명자】
【성명의 국문표기】 임성화
【성명의 영문표기】 LIM,Seong Hwa
【주민등록번호】 751121-1041615
【우편번호】 138-170
【주소】 서울특별시 송파구 송파동 102-9
【국적】 KR
【발명자】
【성명의 국문표기】 박상민
【성명의 영문표기】 PARK,Sang Min
【주민등록번호】 790729-1473915
【우편번호】 343-870
【주소】 충청남도 당진군 대호지면 조금리 182-1
【국적】 KR
【발명자】
【성명의 국문표기】 이수진
【성명의 영문표기】 LEE,Soo Jin
【주민등록번호】 770507-2335211

【우편번호】 138-224
【주소】 서울특별시 송파구 잠실4동 138-501
【국적】 KR
【발명자】
【성명의 국문표기】 이원태
【성명의 영문표기】 LEE,Weon Tae
【주민등록번호】 580815-1041238
【우편번호】 427-070
【주소】 경기도 과천시 주암동 청정드림 201호
【국적】 KR
【취지】 특허법시행규칙 제13조·실용신안법시행규칙 제8조의 규정에의하여 위와 같 이 제출합니다. 대리인 진천웅 (인)
【수수료】
【보정료】 0 원
【기타 수수료】 0 원
【합계】 0 원

【서지사항】

【서류명】 특허출원서
【권리구분】 특허
【수신처】 특허청장
【참조번호】 0002
【제출일자】 2002.10.02
【국제특허분류】 G06F
【발명의 명칭】 생물정보학에서의 데이터베이스 처리 방법
【발명의 영문명칭】 Method to handle database for Bioinformatics
【출원인】
【명칭】 학교법인 대우학원
【출원인코드】 2-1999-901351-3
【대리인】
【성명】 진천웅
【대리인코드】 9-1998-000533-6
【포괄위임등록번호】 2000-055603-1
【발명자】
【성명의 국문표기】 김재훈
【성명의 영문표기】 KIM,JAI Hoon
【주민등록번호】 601208-1025524
【우편번호】 449-843
【주소】 경기도 용인시 수지읍 상현리 만현마을 쌍용 1차 701동 1703호
【국적】 KR
【발명자】
【성명의 국문표기】 김민준
【성명의 영문표기】 KIM,Min Joon
【주민등록번호】 790220-1454734
【우편번호】 442-191
【주소】 경기도 수원시 팔달구 우만1동 519-8 204호
【국적】 KR
【발명자】
【성명의 국문표기】 이성준
【성명의 영문표기】 LEE,Seong Joon

【주민등록번호】 770416-1932122
【우편번호】 690-042
【주소】 제주도 제주시 용담2동 627-15
【국적】 KR

【발명자】
【성명의 국문표기】 임성화
【성명의 영문표기】 LIM,Seong Hwa
【주민등록번호】 751121-1041615
【우편번호】 138-170
【주소】 서울특별시 송파구 송파동 102-9
【국적】 KR

【발명자】
【성명의 국문표기】 박상민
【성명의 영문표기】 PARK,Sang Min
【주민등록번호】 790729-1473915
【우편번호】 343-870
【주소】 충청남도 당진군 대호지면 조금리 182-1
【국적】 KR

【발명자】
【성명의 국문표기】 이수진
【성명의 영문표기】 LEE,Soo Jin
【주민등록번호】 770507-2335211
【우편번호】 138-224
【주소】 서울특별시 송파구 잠실4동 138-501
【국적】 KR

【심사청구】 청구

【취지】 특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사 를 청구합니다. 대리인
진천웅 (인)

【수수료】

【기본출원료】	20	면	29,000 원
【가산출원료】	3	면	3,000 원
【우선권주장료】	0	건	0 원
【심사청구료】	2	항	173,000 원

1020020060295

출력 일자: 2003/7/30

【합계】	205,000 원
【감면사유】	학교
【감면후 수수료】	102,500 원
【첨부서류】	1. 요약서·명세서(도면)_1통

【요약서】

【요약】

본 발명은 생물정보학에서의 데이터베이스 처리 방법에 관한 것으로서, 생물정보학 관련 서열정보를 저장하는 데이터베이스를 처리하는 방법에 관한 것이다.

이러한 본 발명은, 사용자로부터 서비스 대상 서열을 수신하여 큐에 저장하는 제1단계; 제1단계와 병행하여 큐에 서비스 대상 서열이 있는지를 조사하는 제2단계; 큐에 서비스 대상 서열이 있는 경우에는 데이터베이스로부터 현재 순번의 서열을 읽어 큐에 있는 모든 서비스 대상 서열과 비교하는 제3단계; 제3단계에서 비교된 서비스 대상 서열 중 데이터베이스의 모든 서열에 대하여 비교된 것이 있는지를 판단하여, 해당 서비스 대상 서열을 큐로부터 제거하는 제4단계; 및 현재 순번을 하나 증가시키고 제2단계로 진행하는 제5단계를 포함하는 것을 특징으로 한다.

본 발명을 사용하면, 데이터베이스를 한번 액세스해서 현재 처리되는 모든 사용자 요청을 위해 사용하므로 각 사용자 요청에 대해서 데이터베이스를 한번만 액세스한다. 그러므로, 평균 시스템 비용이 감소하고 좋은 응답시간을 갖게 된다. 또한, 종래의 방식에 비하여 같은 하드웨어상에서 받아들일 수 있는 사용자 도착율의 임계값이 높아서 많은 사용자에게 서비스를 제공할 수 있다.

【대표도】

도 3

【색인어】

생물정보학, 서열, 데이터베이스

【명세서】

【발명의 명칭】

생물정보학에서의 데이터베이스 처리 방법{ Method to handle database for Bioinformatics }

【도면의 간단한 설명】

도 1은 종래 방법을 사용하는 경우의 비용에 관한 개요도,

도 2는 본 발명이 적용되는 시스템의 구성도,

도 3은 본 발명의 실시예에 관한 흐름도,

도 4는 구체적인 서비스 방식의 설명을 위한 절차도,

도 5는 시스템 비용에 관한 비교 그래프,

도 6은 응답시간에 관한 비교 그래프를 도시한 것이다.

* 도면의 주요부분에 대한 부호의 설명

1: 사용자 단말 2: 통신 네트워크

3: 서버 3-1: 사용자 요청 접수용 프로그램

3-2: 서열 비교/분석용 프로그램 4: 데이터베이스

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<11> 본 발명은 생물정보학에서의 데이터베이스 처리 방법에 관한 것으로서, 특히 사용자가 생물정보학 관련 서열의 비교를 요청하면 이전에 처리하던 사용자 요청에 대한 처리가 종료될 때까지 기다리지 않고 함께 처리함으로써, 각 사용자 요청에 대해 데이터베이스를 한번만 액세스하도록 하여 시스템 비용과 응답시간에서 이익을 볼 수 있는 방법에 관한 것이다.

<12> 21세기 초에 인간 유전자 프로젝트의 성공적인 수행은 모든 생명과학 분야의 급속한 발전을 야기하였으며, 이러한 인간 유전체 지도의 완성으로 전개되는 유전자 이후시대(Post Genom)에는 인간의 모든 유전자와 유전자의 발현으로 생성되는 단백질들의 구조와 기능에 관한 연구가 활발히 수행될 것이다. 컴퓨터가 0과 1로 표현되는 정보를 저장하고 있듯이, 인간의 유전자는 A, T, G, C라는 네개의 문자로 표현된 약 30억개의 정보를 저장하고 있다. 연구가 진행되면서 막대한 디지털 정보가 축적되고 있으며, 웹(Web)을 통해 공개된 생물정보학 관련 데이터베이스도 SwissProt, GenBank, EMBL 등 매우 많다.

<13> 이러한 생물정보학 관련 데이터베이스를 사용자 요청에 따라 검색하여 비교하고 알맞은 유전자 정보를 찾아주는 다양한 프로그램이 있는데, A, T, G, C로 이루어진 데이터를 비교 검색하여 서열 비교를 수행하는 FastA, Blast, ClustalW 등의 패턴 매치 프로그램과 데이터의 서열로부터 구조를 예측하는 J-NET이나 J-PRED와 같은 프로그램으로 나뉜다.

<14> 미래의 생물학자는 실험보다는 프로그램를 활용한 정보 분석에 더 많은 시간을 투자해야 할 것으로 전망하는 견해가 많다. 유전자 이후 시대의 생물정보학이 단순히 데이터 제공 서비스 이외에 유전자 자체의 완전한 이해를 그 목적으로 하게 되었다는 것을 의미하는 것이다. 이는 프로그램의 더 강력한 기능과 컴퓨팅 파워에 대한 요구의 증대와 연관이 있다. 또한, 생물정보학에서 사용되는 데이터베이스는 연구가 진행됨에 따라 데이터의 크기가 기하급수적으로 커지고 있다. 이런 데이터베이스 크기의 증대는 생물정보학에서 데이터베이스의 효율적인 사용을 더욱 중요하게 부각시키고 있다.

<15> 종래에 사용되고 있는 FastA나 Blast 등의 프로그램들은 웹을 통해 서비스되며, 사용자가 서버에 접속하여 비교하고자 하는 단백질 서열을 전송한다. 그러면 서버는 데이터베이스에서 서열을 읽어들여 사용자가 요청한 서열과 비교한다. 이러한 프로그램들은 데이터베이스 기반으로 작동한다. 즉, 매번 사용자의 요청마다 데이터베이스를 액세스하여 데이터를 읽은 후 사용자의 요구에 응답을 해야 한다. 예로서 FastA의 경우 사용자는 비교/분석하고 싶은 서열을 FastA 서버에 전송한다. 전송되는 사용자 서열은 데이터베이스에 저장되어 있는 각각의 서열과 비교되어 유사도가 검사되고 일정치 이상의 유사도를 갖는 서열을 사용자에게 돌려준다. 이때 서버는 모든 사용자 요청 각각에 대해서 데이터베이스를 액세스한다.

<16> 도 1을 참조하여 이러한 절차를 통해 서비스할 때의 비용을 설명하기로 한다. 여기서 C_{DB}는 사용자 요청이 왔을 때 데이터베이스를 한번 액세스하는 비용이며, C_{seq}는 데이터베이스에서 읽어들인 모든 서열과 사용자가 요청한 서열을 비교 분석하는 비용이다. 즉, 하나의 사용자 요청 Rn(n = 1, 2, 3, ...)에 대해서 서버는 $C_{DB}+C_{seq}$ 만큼의 비용이 소요된다. 이러한 종래의 구조에서는 현재 처리되고 있는 사용자 요청이 없을 때는 사용자 요청이 이루어지면 즉시 사용자 요청을 처리하고, 이미 다른 사용자 요청이 처리되고 있을 때는 새로 발생한 사용자 요

청은 순서대로 큐(Queue)에 등록된다. 도 1에서 요청 R2는 R1이 처리되는 동안 발생 하였기 때문에 R2는 큐에 등록되고, R1의 처리가 모두 끝나는 시점에서 처리됨과 동시에 큐에서 제거된다.

<17> 데이터베이스에서 한 블럭을 읽어들일 때 소요되는 디스크 액세스 시간을 C_{io}, 데이터베이스에 저장되어 있는 전체 서열의 개수를 N_b, 데이터베이스에서 읽어들인 하나의 단백질 서열과 사용자가 요청한 단백질 서열간의 비교 시간, 즉 프로세싱 시간을 C_{cpu}로 정의하기로 한다. 서버는 하나의 사용자 요청 단백질 서열을 비교할 때마다 데이터베이스의 모든 내용을 메모리로 가져와야 한다. 이때 걸리는 시간은 데이터베이스를 한번 액세스하는 시간과 데이터베이스에 저장되어 있는 전체 서열 개수의 곱과 같다. 한 블럭을 읽어들일 때의 시간은 모두 같다고 가정하면 액세스 시간(C_{DB})는 아래의 수학식 1과 같이 나타낼 수 있다.

<18> 【수학식 1】 $C_{DB} = C_{io} \times N_b$

<19> C_{DB}는 데이터베이스의 모든 서열을 액세스하는 시간이며, 데이터베이스 검색을 위한 디스크 액세스 시간이다. 그리고, 각 서열간의 비교 시간은 사용자가 요청한 하나의 서열을 데이터베이스에서 읽은 비교 대상 서열과 비교하는 시간(C_{seq})이 된다. 데이터베이스의 모든 서열과 사용자 요청 서열을 비교하는 시간은 다음의 수학식 2와 같이 나타낼 수 있다.

<20> 【수학식 2】 $C_{seq} = C_{cpu} \times N_b$

<21> 그러면, 한 사용자가 서버에 접속하여 하나의 단백질 서열을 비교하는데 걸리는 평균시간(C^{o}_{avg})은 수학식 1과 수학식 2를 더한 시간으로서 다음의 수학식 3과 같이 나타낼 수 있다.

<22> 【수학식 3】 $C^{o}_{avg} = C_{DB} + C_{seq} = C_{io} N_b + C_{cpu} N_b = (C_{io} + C_{cpu}) N_b$

<23> 종래의 방법에 대한 응답시간을 설명하기로 한다. 이 때, 사용자 요청은 발생율 λ의 포아송과정(Poisson process)이라 가정하기로 한다. 서버가 하나의 사용자 요청을 처리하고 있을 때 다른 사용자 요청이 발생하면 새로운 사용자 요청은 큐에 등록된다. 즉, 사용자 요청들은 발생한 순서대로 큐에 등록되고, 큐에 등록된 순서대로 순차적으로 서비스된다. 모든 요청의 서비스 비용이 같다고 가정하면 M/G/1 큐잉 모델이 된다.

<24> 서비스 시간 $1/\mu$은 단일 사용자 요청을 처리하는 시간과 같다. 즉, 서비스 시간 $1/\mu$은 하나의 사용자 요청이 서비스를 받는 평균비용(C^{o}_{avg})이 된다. 여기서 서비스율 μ는 $\frac{1}{C^{o}_{avg}}$ 로 표시된다. 사용자 요청 발생율(λ)과 서비스율(μ)을 M/G/1 큐잉 모델의 응답시간에 대입해 본 결과는 다음의 수학식 4와 같다.

<25>

【수학식 4】

$$W_o = \left(\frac{1}{C^{o}_{avg}}\right)^{-1} + \frac{\lambda \cdot \left(\frac{1}{C^{o}_{avg}}\right)^{-2}}{2\left(1-\frac{\lambda}{1/C^{o}_{avg}}\right)} = C^{o}_{avg} + \frac{\lambda C^{o}_{avg}{}^{2}}{2(1-\lambda \cdot C^{o}_{avg})}$$

<26> 위에서 설명한 바와 같이, 종래의 방법을 사용하면 각 사용자 요청에 대하여 매번 데이터베이스의 검색을 수행해야 하므로 많은 시스템 비용이 소요된다. 또한 서버에 과부하를 초래하여 응답시간이 길어질 수 있다.

【발명이 이루고자 하는 기술적 과제】

<27> 본 발명은 상기와 같은 문제점을 해결하기 위하여 제안된 것으로서, 사용자가 생물정보학 관련 서열의 비교를 요청하면 이전에 처리하던 사용자 요청에 대한 처리가 종료될 때까지

기다리지 않고 함께 처리함으로써, 각 사용자 요청에 대해 데이터베이스를 한번만 액세스하도록 하여 시스템 비용과 응답시간에서 이익을 볼 수 있는 방법을 제공하는데 그 목적이 있다.

<28> 상기와 같은 목적을 달성하기 위하여, 본 발명에 따른 생물정보학에서의 데이터베이스 처리 방법은, 생물정보학 관련 서열정보를 저장하는 데이터베이스와 연동하고 일정 통신 네트워크를 통해 각 사용자 단말과 접속하는 서버에서, 상기 각 사용자 단말로부터 요청된 서비스 대상 서열을 상기 데이터베이스의 서열과 비교/분석하기 위하여 상기 데이터베이스를 처리하는 방법에 있어서, 상기 사용자 단말로부터 서비스 대상 서열을 수신하여 큐(Queue)에 저장하는 제1 단계; 상기 제1 단계와 병행하여, 상기 큐에 서비스 대상 서열이 있는지를 조사하는 제2 단계; 상기 제2 단계에서의 조사 결과, 상기 큐에 서비스 대상 서열이 있는 경우에는 상기 데이터베이스로부터 현재 순번의 서열을 읽어 상기 큐에 있는 모든 서비스 대상 서열과 비교/분석하는 제3 단계; 상기 제3 단계에서 비교/분석된 서비스 대상 서열 중 상기 데이터베이스의 모든 서열에 대하여 비교/분석된 것이 있는지를 판단하여, 해당 서비스 대상 서열을 상기 큐로부터 제거하는 제4 단계; 및 상기 현재 순번을 하나 증가시키되, 상기 데이터베이스의 모든 서열을 읽은 경우에는 상기 현재 순번을 초기화하고, 상기 제2 단계로 진행하는 제5 단계를 포함하는 것을 특징으로 한다.

【발명의 구성 및 작용】

<29> 이하, 첨부된 도면을 참조하여 본 발명을 상세히 설명하기로 한다.

<30> 도 2를 참조하여 본 발명이 적용되는 시스템의 개요를 설명하자면, 본 발명에 따라 생물정보학 관련 서열정보에 대한 비교/분석 서비스를 제공하는 주체는 서버(3)이다.

<31> 이 서버(3)는 생물정보학 관련 서열정보를 저장하는 데이터베이스(4)와 연동하고, 일정 통신 네트워크(2)를 통해 각 사용자 단말(1:클라이언트)과 접속한다. 여기서 통신 네트워크(2)는 인터넷망인 것이 바람직하다. 본 발명은 서버(3)에 설치되는 사용자 요청 접수용 프로그램(3-1)과 서열 비교/분석용 프로그램(3-2)에 의하여 바람직하게 구현될 수 있다.

<32> 각 사용자들은 자신이 비교하기 원하는 서열정보를 서버(3)로 전송하여 서열의 비교/분석을 요청하며, 서버(3)는 사용자가 요청한 서열을 데이터베이스(4)에 저장되어 있는 서열정보와 비교/분석하여 그 결과를 해당 사용자 단말로 보내준다.

<33> 본 발명은 데이터베이스(4)를 액세스하는 방법에 그 핵심이 있는 것이며, 서버(3)에서 수행하는 비교 및 분석 방법은 종래에 각 서버에서 이루어지고 있는 방법과 동일한 것이므로 비교 및 분석과 관련한 상세 설명은 생략하기로 한다.

<34> 도 3을 참조하여 데이터베이스에 저장되어 있는 서열이 D(n)(n=1,2,3,...,n)인 경우에 대한 바람직한 실시예를 설명하기로 한다.

<35> 사용자 요청 접수용 프로그램(3-1)은 사용자 단말(1)로부터의 서비스 요청을 대기하고 있다가(S11), 사용자 요청이 발생하면 사용자가 요청한 서비스 대상 서열을 수신하여 큐(Queue)에 저장한다(S12,S13)(제1단계).

<36> 서열 비교/분석용 프로그램(3-2)은 제1 단계(S11 내지 S13)와 병행하여 큐에 서비스 대상 서열(사용자 요청)이 있는지를 조사하고 있다(S21:제2단계). 즉, 사용자 요청 접수용 프로그램(3-1)은 서열 비교/분석용 프로그램(3-2)과 병행하여 동작하는 것이며, 큐를 통해 서비스 대상 서열을 주고 받으며 서로 독립적으로 동작한다.

<37> 한편, 서열 비교/분석용 프로그램(3-2)은 그 동작과 관련하여 일정 변수(k)를 초기화하여 설정하고 있으며, 제2 단계(S21)에서의 조사 결과 큐에 서비스 대상 서열이 있는 경우에는 데이터베이스(4)로부터 k번째 서열을 읽은 후, 큐에 있는 모든 서비스 대상 서열과 비교/분석한다(S22,S23:제3단계).

<38> 그리고, 서열 비교/분석용 프로그램(3-2)은 제3 단계에서 비교/분석된 서비스 대상 서열 중 데이터베이스(4)의 모든 서열(D(1) 내지 D(n))에 대하여 비교/분석된 것이 있는지를 판단하여, 그러한 서비스 대상 서열이 있으면 이를 큐로부터 제거한다(S24,S25:제4단계). 즉, 비교가 종료한 서열을 큐에서 제거하는 것이다. 제4 단계를 진행한 후에는 $k \neq n$인 경우에는 k를 하나 증가시키되, k = n인 경우에는 k를 초기화하고 단계 S21로 되돌아간다(S26 내지 S28:제5단계).

<39> 이와 같이 본 발명은 현재 요청된 서비스 대상 서열을 이미 요청되어 처리되고 있던 서비스 대상 서열과 함께 처리한다. 이것은 생물정보학의 서열 검색에 있어서는 데이터베이스의 모든 서열을 전부 탐색하는 것이 일반적이고, 생명정보학의 데이터베이스는 각각의 데이터들 사이에 의존성이 없으므로, 데이터들이 처리되는 순서와 상관없이 모든 데이터를 처리할 수 있다는 특징이 있기 때문에 가능한 것이다.

<40> 도 4를 참조하여, 4개의 사용자 요청(Rn, n=1,2,3,4)에 대하여 서비스가 이루어지는 실시예를 설명하기로 한다. 여기서 D(i)는 i번째 데이터베이스의 서열을 액세스하는데 소비되는

비용이며, 데이터베이스는 4개의 서열만을 갖는다고 가정한다. 그리고, R(i,j)는 i번째 사용자 요청을 처리하기 위해서 j번째 데이터베이스 서열과 비교하는데 소요되는 비용을 나타낸다.

<41> 첫번째 서열 D(1)을 읽고 사용자 요청 R1을 위한 서비스 R(1,1)을 처리하게 된다. 서비스 R(1,1)을 처리하는 동안 사용자 요청 R2가 발생하고, 두번째 서열 D(2)를 읽고 R1을 위한 서비스 R(1,2)와 R2를 위한 서비스 R(2,2)를 처리하게 된다. 즉, 데이터베이스는 한번만 액세스하고 여러 사용자 요청에 대해서 처리하므로 데이터베이스를 액세스하는 회수를 줄일 수 있다. 서비스 요청 R1은 4번째 서열 D(4)까지 읽은 후 서비스를 마치게 된다. 4번째 서열 D4까지 읽은 후 요청 R2는 첫번째 서열 D1을 위한 처리를 하지 않았으므로 사용자 요청 R3과 함께 첫번째 서열 D1을 읽고 처리하는 루틴을 실행하게 된다. 즉, 사용자 요청은 데이터베이스의 모든 서열을 액세스할 때까지 처리하게 되지만, 이전에 처리되고 있던 사용자 요청에 대한 처리가 끝날 때까지 새로운 요청이 지연되지 않는다는 장점이 있다. 이는 생물정보학의 서열 비교에 있어서는 데이터베이스에서 데이터를 읽는 순서가 결과에 미치는 영향이 없기 때문에 가능하다.

<42> 본 발명에 따라 서비스를 제공할 때의 비용은 데이터베이스의 시작점으로부터 액세스를 시작하여 다시 시작점으로 돌아오기 전까지의 처리시간을 기준으로 구할 수 있다. 사용자 요청의 발생율을 λ라 가정하고 데이터베이스를 모두 액세스하는 동안 소요되는 데이터베이스 액세스 시간과 도착한 사용자 요청과 비교하는 시간의 합을 C_{total}^{cp}라 하자. 이 주기(C_{total}^{cp}) 동안 발생하는 평균 요청의 수는 $\lambda \cdot C_{total}^{cp}$ 가 된다. 이 주기(C_{total}^{cp}) 동안 데이터베이스는 한번만 액세스되며, 주기 동안의 총 비용을 구해보면 다음의 식 5와 같다.

<43> 【수학식 5】 $$C_{total}^{cp} = C_{DB} + \lambda \cdot C_{total}^{CP} \cdot C_{seq}$$

<44> 수학식 5를 C_{total}^{cp}에 대하여 정리하면 다음의 수학식 6과 같다.

<45> 【수학식 6】 $$C_{total}^{CP} = \frac{C_{DB}}{1-\lambda \cdot C_{seq}}$$

<46> 하나의 요청도 발생하지 않는 경우를 고려해서 하나의 사용자 요청을 처리하는데 소요되는 시스템 비용을 구해보면 다음의 수학식 7과 같이 나타낼 수 있다.

<47> 【수학식 7】 $$C_{avg}^{cp} = \frac{C_{DB}}{C_{total}^{cp} \cdot \lambda}(1 - e^{-\lambda \frac{C_{DB}}{1-\lambda C_{seq}}}) + C_{seq}$$

<48> 수학식 7은 수학식 5를 사용자 요청의 수로 나눈 값, 즉 한 주기에 사용자 요청이 발생할 확율($1 - e^{-\lambda C_{total}^{cp}}$)을 고려해서 하나의 사용자 요청이 처리되는 비용을 구한 것이다.

<49> 수학식 7을 정리하면 다음의 수학식 8과 같다.

<50> 【수학식 8】 $$C_{avg}^{cp} = (\frac{1}{\lambda} - C_{seq})(1 - e^{-\lambda \frac{C_{DB}}{1-\lambda C_{seq}}}) + C_{seq}$$

<51> 본 발명에 따라 서비스를 제공할 때의 응답시간을 구해보기로 한다.

<52> 하나의 사용자 요청에 대하여 서비스가 완료되는 시점은 데이터베이스의 모든 서열을 읽고 이에 대한 처리가 모두 끝나는 때이다. 각각 읽혀진 데이터베이스의 서열은 동시에 처리되고 있는 다른 사용자 요청을 위해서도 사용된다. 즉, 본 발명에서의 응답시간은 하나의 사용자 요청을 처리하는 시간과, 이를 처리하는 시간($C_{DB} + C_{seq}$) 동안 함께 처리되는 사용자 요청의 처리시간($\lambda \cdot W_{cp} \cdot C_{seq}$)의 합으로서 다음의 수학식 9와 같이 나타낼 수 있다.

<53> 【수학식 9】 $$W_{cp} = C_{DB} + C_{seq} + \lambda \cdot W_{cp} \cdot C_{seq}$$

<54> 수학식 9를 정리하면 다음의 수학식 10과 같다.

<55>

【수학식 10】 $$W_{cp}=\frac{C_{DB}+C_{seq}}{1-\lambda \cdot C_{seq}}$$

<56> 이제 종래의 방법과 본 발명을 이용한 경우에 대하여 성능을 비교해보기로 한다.

<57> 각 방법은 사용자 요청의 도착율 λ에 대해서 임계값을 갖는데, 도착율 λ의 임계값은 서버의 사용율이 1보다 적은 조건을 만족하는 최대의 도착율를 나타낸다. 서버의 사용율은 사용자 요청의 도착율과 서버가 각 방식을 사용해서 하나의 사용자 요청을 처리하는 평균비용의 곱으로 나타낼 수 있다. 이 값이 1보다 작을 경우 서비스가 가능한 것이다. 물론 데이터베이스 액세스와 CPU 사용을 동시에 할 수 있는 기법을 이용하면 서버의 사용율을 1보다 높일 수 있다.

<58> 서열의 비교를 순차적으로 수행하는 방식을 가정하고, 각 방식에서의 임계값을 구해보면 다음과 같다.

<59> 1) 종래 방식

<60> 종래의 방식에서 하나의 사용자 요청을 처리하는 평균비용은 수학식 3과 같이 나타낼 수 있다. 이를 식으로 나타내면 $\lambda \cdot C^{o}_{avg} < 1$과 같다. 즉, 종래의 방식에서 λ의 임계값은 다음의 수학식 11과 같이 나타낼 수 있다.

<61>

【수학식 11】 $$\lambda \leq \frac{1}{C_{seq}+C_{DB}}$$

<62> 2) 본 발명에 따른 방식

<63> 본 발명에 따를 경우 서버의 사용율은 $\lambda \cdot \frac{1}{\lambda}$ 이 되므로 항상 조건을 만족한다. 또한 수학식 9에서 C_{total}^{cp}는 양수이고 C_{DB} 또한 양수이다. 즉, $1-\lambda \cdot C_{seq} > 0$을 만족해야 한다. 이를 풀면 다음의 수학식 12와 같이 λ의 임계값을 얻을 수 있다.

<64> 【수학식 12】 $$\lambda < \frac{1}{C_{seq}}$$

<65> 생물정보학에서 실제로 사용되고 있는 데이터베이스 GenBank(Protein Sequence Database of Rip International Release 72.02)는 1981년 미국립보건원으로부터 지원을 받아 로스 알라모스 연구소가 이를 관리하다가 1992년 미국립보건원의 국립의학도서관 산하 미국립생물공학정보센터(NCBI)로 이전되어 관리되는 서열정보 데이터베이스이다. 이러한 GenBank에 대하여 사용자가 요청하는 서열로서 인간(human) 단백질 중 세포의 산화 환원에 작용하는 색소 단백질(cytochrome)을 사용해 보았다. 그 결과 C_{DB}는 3.99 sec가 되고, 데이터베이스의 모든 서열과 사용자 요청 서열을 모두 비교하는 비용, C_{seq}는 19.98 sec가 되었다.

<66> 수학식 11과 수학식 12를 비교하면 수학식 12에서의 λ의 최대값은 수학식 11에서의 λ의 최대값보다 항상 큰 값을 갖는다. 즉, 같은 성능의 하드웨어에서 본 발명에 따른 방식을 사용하면 종래의 방식을 사용하는 경우보다 더 많은 사용자를 받을 수 있음을 알 수 있다.

<67> 도 5를 참조하여, 종래의 방식과 본 발명에 따른 방식을 이용할 때의 시스템 비용을 수학식 3과 수학식 8을 이용하여 비교해보기로 한다. 각 방식에 대하여 임계값까지 그래프에 표시하였으며, x축은 사용자 요청율 λ이고 y축은 사용자당 시스템 비용을 나타낸다.

<68> 본 발명에 따른 방식을 사용하면, λ값이 증가할 수록 시스템 비용이 줄어드는 것을 볼 수 있는데, 사용자 요청율 λ가 커질수록 상대적인 데이터베이스의 액세스 비용이 줄어들기 때문에 사용자당 비용이 급격히 감소하게 된다.

<69> 도 6을 참조하여 종래의 방식과 본 발명에 따른 방식의 응답시간을 비교하기로 한다. 여기서 x축은 사용자 요청율 λ를 나타내고 y축은 임의의 사용자 요청에 대한 프로그램의 평균응답시간을 나타낸다.

<70> 종래의 방식은 임계점에 가까워질수록 응답시간이 급격히 증가하지만, 본 발명에 따른 방식을 사용하면 짧은 응답시간을 보였다. 이는 사용자가 적을 때는 데이터베이스를 즉시 읽기 때문에 응답이 빠를 수 있는 것이다. 또한, 종래 방식보다 데이터베이스를 액세스하는 회수가 적기 때문에 응답시간이 기존방식보다 더 좋은 것이다.

<71> 도 3을 통해 설명한 바와 같이 본 발명에 따른 데이터베이스 처리 방법은 서버(3)에서 수행되는 프로그램에 의하여 바람직하게 구현될 수 있는 것이다. 그러므로, 본 발명은 상기 제1 단계 내지 제5 단계를 수행할 수 있는 컴퓨터 프로그램을 기록한 기록매체도 그 대상으로 한다.

【발명의 효과】

<72> 본 발명을 사용하면, 데이터베이스를 한번 액세스해서 현재 처리되는 모든 사용자 요청을 위해 사용하므로, 각 사용자 요청에 대해서 데이터베이스를 한번만 액세스한다. 그러므로, 평균 시스템 비용이 감소하고 좋은 응답시간을 갖게 된다. 또한, 종래의 방식에 비하여 같은

하드웨어상에서 받아들일 수 있는 사용자 도착율의 임계값이 높아서 많은 사용자에게 서비스를 제공할 수 있다.

【특허청구범위】

【청구항 1】

생물정보학 관련 서열정보를 저장하는 데이터베이스와 연동하고 일정 통신 네트워크를 통해 각 사용자 단말과 접속하는 서버에서, 상기 각 사용자 단말로부터 요청된 서비스 대상 서열을 상기 데이터베이스의 서열과 비교/분석하기 위하여 상기 데이터베이스를 처리하는 방법에 있어서,

상기 사용자 단말로부터 서비스 대상 서열을 수신하여 큐(Queue)에 저장하는 제1 단계;

상기 제1 단계와 병행하여, 상기 큐에 서비스 대상 서열이 있는지를 조사하는 제2 단계;

상기 제2 단계에서의 조사 결과, 상기 큐에 서비스 대상 서열이 있는 경우에는 상기 데이터베이스로부터 현재 순번의 서열을 읽어 상기 큐에 있는 모든 서비스 대상 서열과 비교/분석하는 제3 단계;

상기 제3 단계에서 비교/분석된 서비스 대상 서열 중 상기 데이터베이스의 모든 서열에 대하여 비교/분석된 것이 있는지를 판단하여, 해당 서비스 대상 서열을 상기 큐로부터 제거하는 제4 단계; 및

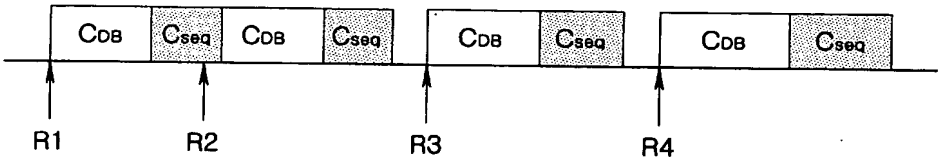
상기 현재 순번을 하나 증가시키되, 상기 데이터베이스의 모든 서열을 읽은 경우에는 상기 현재 순번을 초기화하고, 상기 제2 단계로 진행하는 제5 단계를 포함하는 것을 특징으로 하는 생물정보학에서의 데이터베이스 처리 방법.

【청구항 2】

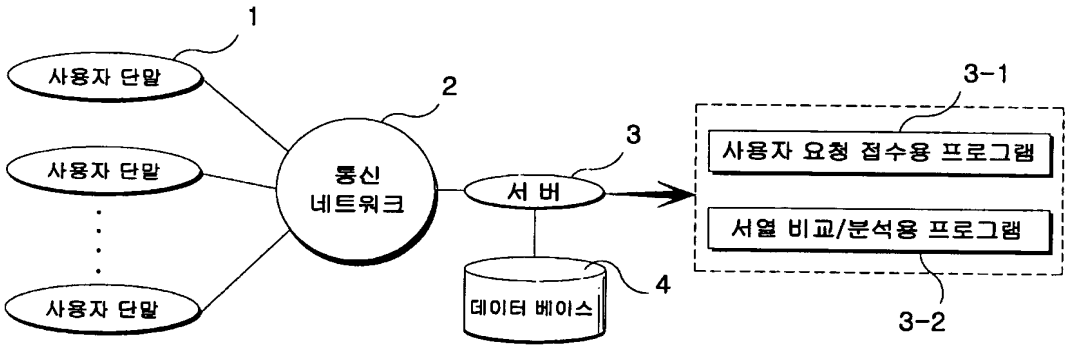
상기 제 1 항에 기재된 제1 단계 내지 제5 단계를 수행하기 위한 컴퓨터 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.

【도면】

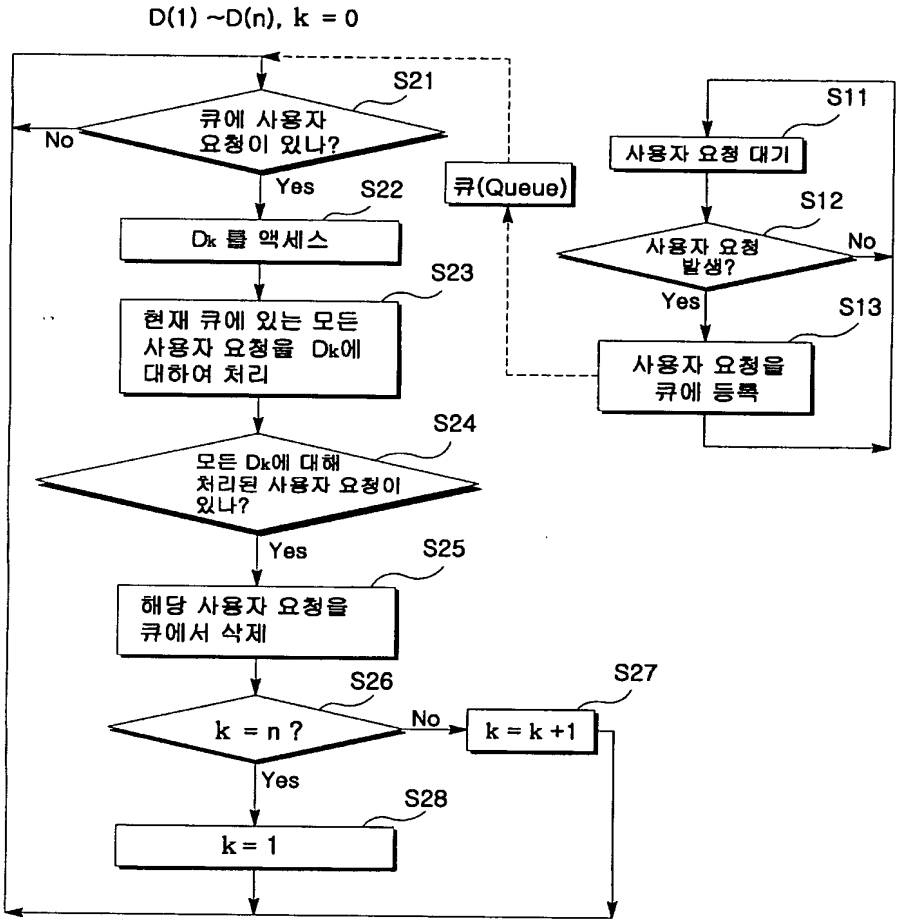
【도 1】


CDB
Cseq
CDB
Cseq
CDB
Cseq
CDB
Cseq
R1
R2
R3
R4

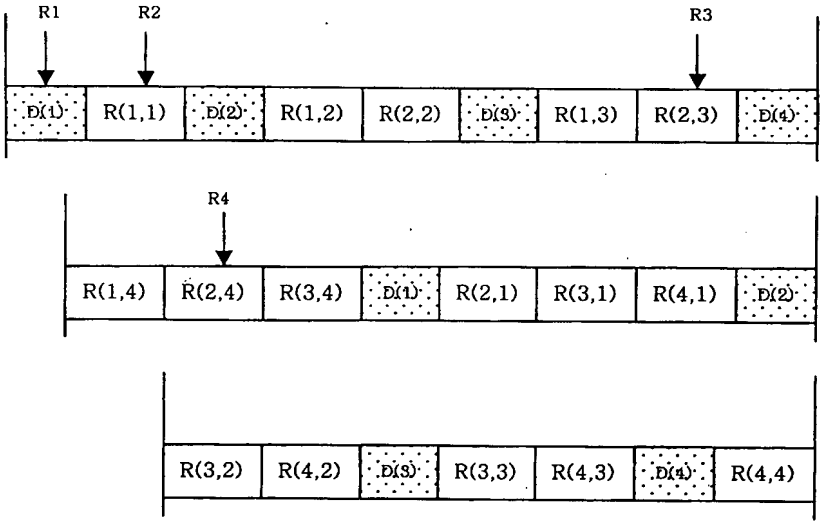
【도 2】


1
사용자 단말
사용자 단말
사용자 단말
2
통신
네트워크
3
서 버
4
데이터 베이스
3-1
사용자 요청 접수용 프로그램
서열 비교/분석용 프로그램
3-2

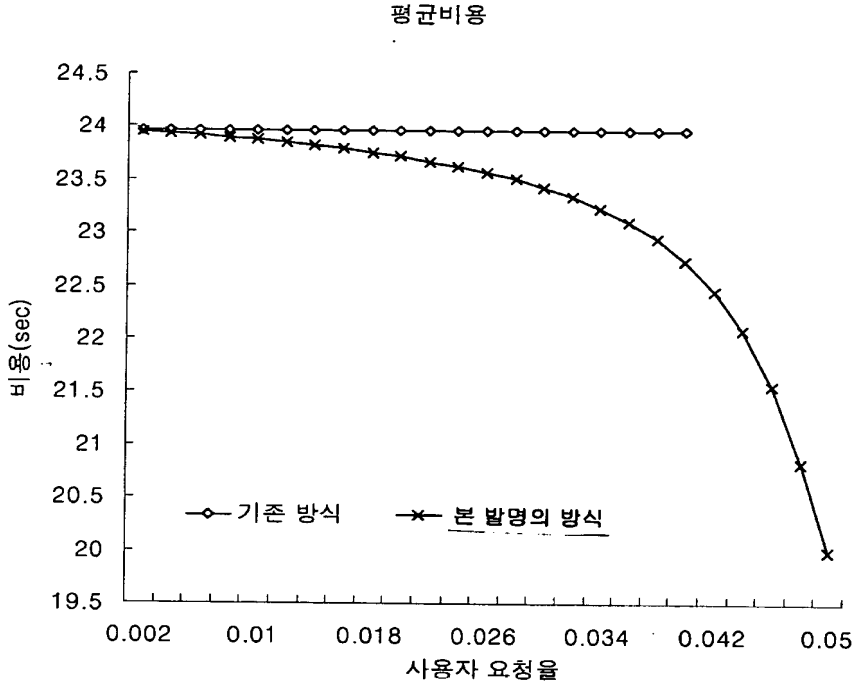
【도 3】


D(1) ~D(n), k = 0
S21
큐에 사용자 요청이 있나?
No
Yes
S22
Dk 를 액세스
S23
현재 큐에 있는 모든 사용자 요청을 Dk에 대하여 처리
S24
모든 Dk에 대해 처리된 사용자 요청이 있나?
Yes
S25
해당 사용자 요청을 큐에서 삭제
S26
k = n ?
No
S27
k = k +1
Yes
S28
k = 1
큐(Queue)
S11
사용자 요청 대기
S12
사용자 요청 발생?
No
Yes
S13
사용자 요청을 큐에 등록

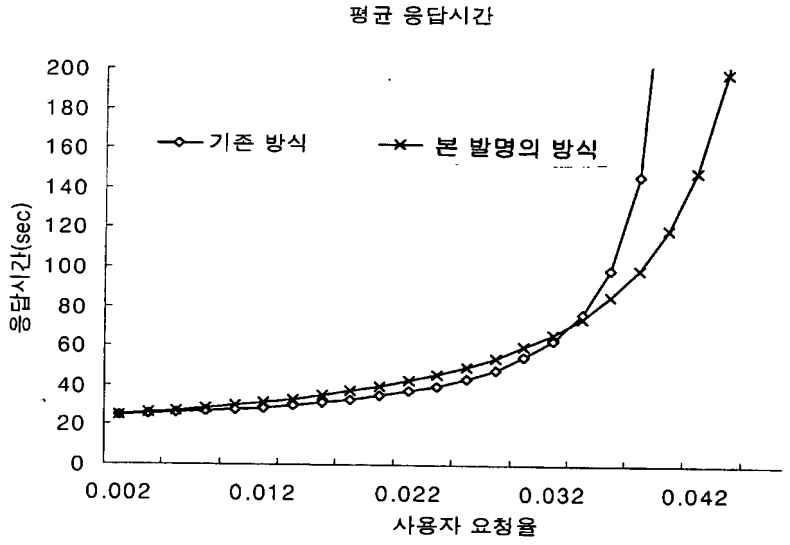
【도 4】


R1
R2
R3
D(1)
R(1,1)
D(2)
R(1,2)
R(2,2)
D(3)
R(1,3)
R(2,3)
D(4)
R4
R(1,4)
R(2,4)
R(3,4)
D(1)
R(2,1)
R(3,1)
R(4,1)
D(2)
R(3,2)
R(4,2)
D(3)
R(3,3)
R(4,3)
D(4)
R(4,4)

【도 5】


평균비용
24.5
24
23.5
23
22.5
22
21.5
21
20.5
20
19.5
비용(sec)
기존 방식
본 발명의 방식
0.002
0.01
0.018
0.026
0.034
0.042
0.05
사용자 요청율

【도 6】


평균 응답시간
200
180
160
140
120
100
80
60
40
20
0
응답시간(sec)
기존 방식
본 발명의 방식
0.002
0.012
0.022
0.032
0.042
사용자 요청율